Input Variable Selection for Non-Parametric Regression, Classification, and Time Series Modeling

N0014-96-1-0476 – Productivity Report February 1996 through June 1997

John E. Moody PI
Oregon Graduate Institute
Department of Computer Science and Engineering
P.O. Box 91000, Portland, OR 97291-1000
moody@cse.ogi.edu

Introduction

Variable selection is a critical step in constructing statistical regression, pattern classification, or time series models that are capable of optimum generalization performance. Since the project got started in February 1996, we have implemented the prototype K-test as proposed, carried out extensive testing on regression and time series problems, and developed a selection criterion based upon unsupervised clustering methods. The latter can be applied to both regression and classification type problems.

Under ONR sponsorship, a number of criterion functions have been devised and tested for developing the variable selection methodologies. The work on this project has been conducted by Hong Pi and John Moody. Since Hong Pi has taken a job in industry, Howard Yang (from Amari's research group in Tokyo) will continue working on the project in place of Hong.

Input Selection for Non-Parametric Regression, Classification, and Time Series Modeling - Status Report

Generally, an input variable selection algorithm consists of two parts:

A. A criterion function measuring the optimality of subsets of variables.

B. A search algorithm that searches through the space of all possible subsets of variables and finds the "best" subset, based on the criterion function in A.

A number of criterion functions have been devised and tested for developing the variable selection methodologies.

Estimate of Residual Variance

The K-Test

The K-statistic averages over the variances of a "local neighborhood" chosen from the K nearest neighbors.

$$\begin{aligned}
\hat{\sigma}_y^2(\Delta_{KNN}) &= \langle\langle(y-\overline{y})^2\rangle_{KNN}\rangle \\
&= \frac{1}{2}\langle\langle(y-y')^2\rangle_{KNN}\rangle \\
&\approx \frac{1}{2}\langle\langle(f(\mathbf{x})-f(\mathbf{x}'))^2\rangle_{KNN}\rangle + \frac{1}{2}\langle\langle(r-r')^2\rangle_{KNN}\rangle \\
&= \frac{1}{2}\langle\langle(f'(\mathbf{x})\cdot(\mathbf{x}'-\mathbf{x}))^2\rangle_{KNN}\rangle + \langle\hat{\sigma}_r^2\rangle \\
&\approx \hat{\sigma}_r^2 + \beta\cdot\Delta_{KNN} \qquad (1)
\end{aligned}$$

where

$$\Delta_{KNN} = \langle\langle(\mathbf{x}'-\mathbf{x})^2\rangle_{KNN}\rangle \qquad (2)$$

A linear extrapolation is found to be useful to improve the variance estimate.

Linear extrapolation

Linearly fitting $\hat{\sigma}_y^2(\Delta_\delta)$ and extrapolating, the intercept gives a variance estimate

$$\langle \hat{\sigma}_r^2 \rangle = \hat{\sigma}_y^2(\Delta_\delta = 0) \tag{3}$$

Some numerical results

The various methods for variance estimate are tested on a artificial data set of the type

$$y = \sum_{i=1}^{n_{inp}} \sin(i\pi x_i + \phi_i) + r \tag{4}$$

Figure 1 shows a 2-inputs example. In this case all methods give good estimates on the noise variance.

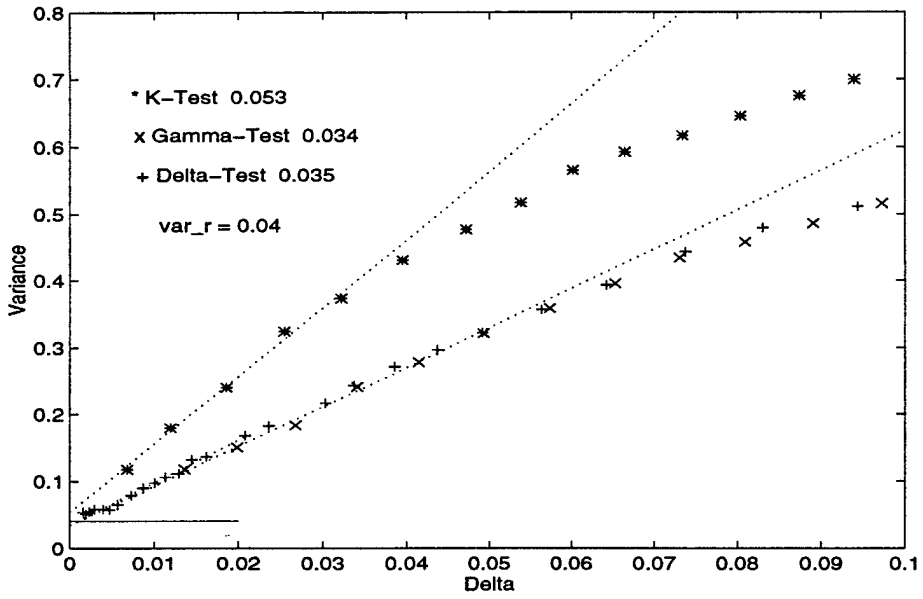

Figure 1: The variance statistics versus the Delta. The data set used features $n_{inp} = 2$, $N = 300$, $\sigma_r^2 = 0.04$ (marked by the horizontal line).

When the input dimension is increased to three, as shown in figure 2, the K-nearest neighbor based methods give misleading results. The *delta*-test does well probing the small Δ region. However if a linear extrapolation is used it over-shoots the target variance.

It is found that

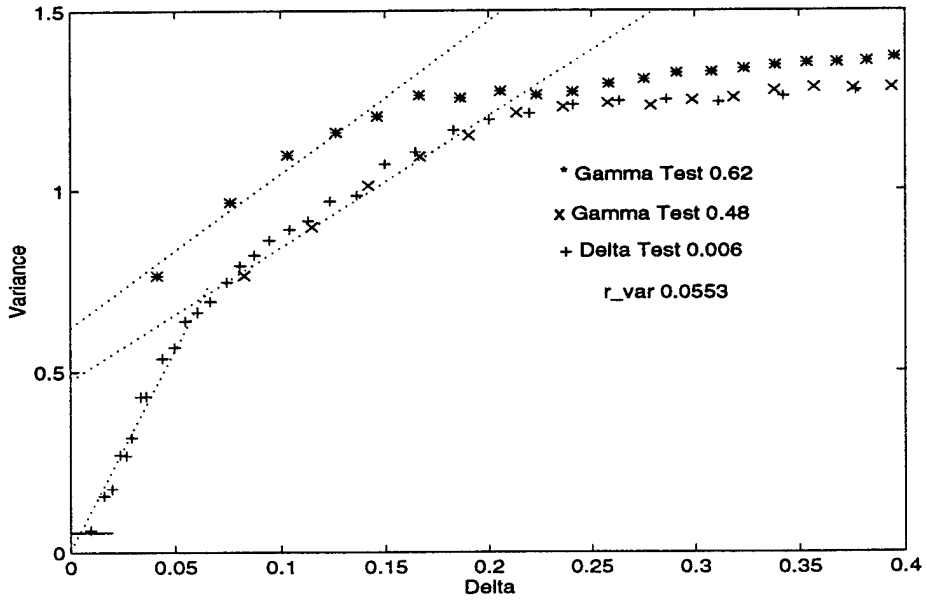

Figure 2: The variance statistics versus the Delta. The data set used features $n_{inp} = 3$, $N = 400$, $\sigma_r^2 = 0.055$ (marked by the horizontal line).

- For variance estimate, the region of $\Delta \to 0$ is crucial.
- The linear extrapolation is useful but can be very unreliable. Further research is needed to improve the reliability.
- It is possible to alter the K-test somewhat by averaging over the local neighborhoods of like "sizes" rather than like K's. This approach might yield better results.

Variable Sensitivity Estimate

Adding noise to one of the variables, its effect on variance estimate gives a measure on variable sensitivity.

$$\xi_i = x_i + \epsilon \tag{5}$$

$$y = f(\mathbf{x}) + r = f(\mathbf{x}_\xi) - \frac{\partial f}{\partial x_i}\epsilon + r \tag{6}$$

where $\mathbf{x}_\xi$ denotes the set of variables $\{x_1, x_2 ... x_{i-1}, \xi_i, x_{i+1} ... x_D\}$.

The variance estimate

$$\begin{aligned}
\hat{\sigma}_y^2(\Delta_\delta) &= \frac{1}{2}\langle (y-y')^2 |\, |\mathbf{x}_\xi - \mathbf{x}'| \le \delta \rangle \\
&= \frac{1}{2}\langle \left[f(\mathbf{x}_\xi) - \frac{\partial f}{\partial x_i}\epsilon + r - (f(\mathbf{x}'+r') \right]^2 \rangle_\delta \\
&= \frac{1}{2}\langle (f(\mathbf{x}_\xi) - f(\mathbf{x}'))^2 \rangle_\delta + \frac{1}{2}\langle \left(\frac{\partial f}{\partial x_i}\right)^2 \rangle \overline{\epsilon^2} + \frac{1}{2}\langle (r-r')^2 \rangle_\delta \\
&= \hat{\sigma}_r^2 + \frac{1}{2}\langle \left(\frac{\partial f}{\partial x_i}\right)^2 \rangle_\delta \overline{\epsilon^2} + \beta \cdot \Delta_\delta \qquad (7)
\end{aligned}$$

The linear extrapolation results in

$$\hat{\sigma}_y^2(\Delta_\delta = 0) = \langle \hat{\sigma}_r^2 \rangle + \frac{1}{2}\langle \left(\frac{\partial f}{\partial x_i}\right)^2 \rangle_\delta \overline{\epsilon^2} \qquad (8)$$

Fitting two values $\overline{\epsilon_1^2}$ and $\overline{\epsilon_2^2}$ produce readings on both $\langle \hat{\sigma}_r^2 \rangle$ and $S_i \equiv \frac{1}{2}\langle \left(\frac{\partial f}{\partial x_i}\right)^2 \rangle_\delta$, where the slope S_i provides a sensitivity measure on the i:th input.

Taking $\epsilon \to -\epsilon$ helps reducing some of the biases.

Selection Criterion Based on Clustering

The variance estimates are applicable only to regression problems. An alternative method is developed based on an unsupervised clustering algorithm (Ball 1965, Therrien 1989). By examining the characteristics of the clusters formed, suitable criterion functions can be defined for both classification and regression problems. Search algorithms can then be applied to find the "best" subset of input variables.

Criterion Function

Given a data set $\{y, \mathbf{x}\}$ where y is the dependent variable, and $\mathbf{x} = \{x_1, x_2, ..., x_D\}$ are the independent variables, the criterion function should be so defined that it provides a measure of how well y can be represented by a mapping from $\mathbf{x}$. This requirement applies to either regression problems where y is a continuous variable, or for classification problem where y is a discrete set of class labels.

The construction of a criterion function is to be based on the following observations: If a one-to-one mapping exists between $\mathbf{x}$ and y, then a set of points forming a cluster in the $\mathbf{x}$ space should also resemble a cluster in the y space. This leads to that, for regression, the variance along the y axis will be small for the set of points on this cluster; and for classification, the cluster should overwhelmingly represented by samples from a particular class.

Based on this observation, a criterion function can be defined in the following manner:

- Assign data vectors into clusters in the $\mathbf{x}$ space. This can be achieved by a clustering algorithm. In the current approach, the ISODATA algorithm, which is a variation of the K-means algorithm with heuristics for cluster splitting and merging, is adopted.
- For classification problems, define the criterion function as

$$C(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{K} n_{max}(i) \tag{9}$$

  where N is the total number of points, K is the number of clusters, and $n_{max}(i)$ is the number of points in the class that are most represented on the i:th cluster. The ideal situation would be that the points on a cluster all have the same class label, in which case $C(\mathbf{x}) = 1$.

  For regression problems, the criterion is

$$C_r(\mathbf{x}) = 1 - \frac{v(\mathbf{x})}{\sigma_y^2} \tag{10}$$

  where σ_y^2 is the variance along the y direction of all the points, and $v(\mathbf{x})$ is a local variance measure defined by

$$v(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{K} n(i)\left(\frac{1}{n(i)-1}\sum_{j=1}^{n(i)}(y_j - \overline{y}_j)^2\right) \tag{11}$$

  Here $n(i)$ is the number of points on the i:th cluster, and the $(\cdot)$ is the y variance over this set of points. $v(\mathbf{x})$ is the y variance of a cluster averaged over all clusters. C_r is defined in analogous to the R^2 measure in the linear regression theory. Clearly C_r approaches to unity when the input-output relationship is captured perfectly by the clustering mechanism.

Search Methods

An exhaustive search over all possible variable subsets would require $O(2^D)$ evaluations of the criterion function. This is possible practically only when the number of variables D

is small. When D is larger than say 15, more efficient search algorithms must be utilized. Forward selection and backward elimination algorithms (see e.g. Miller 1990) have been implemented.

A forward selection algorithm starts from a null set and add a variable one by one as long as it increases the value of the criterion function. This method is extremely simple and efficient. Because it starts from a low dimension space, it generally produces quite reliable selections if the number of good variables is small comparing to the total number of candidate variables. The ordering of the variables can become a problem though. The algorithm tends to pick up variables placed near the front and ignore the variables near the end.

The backward elimination algorithm starts from a full variable set. A variable is then eliminated if its removal does not cause a significant deterioration in the value of the criterion function. Specifically, the algorithm consists of the following steps:

Step 1: With all D candidate variables included, calculate the corresponding criterion function value CR_0.

Step 2: For $i = D, D-1, ...1$, construct a variable subset consisting of the original set of variables excluding the i:th one, compute its corresponding criterion value CR_i; if $CR_0 - CR_i \leq 0$, define $i_r = i$ and break the loop. The i_r:th variable is a target for removal.

Step 3: If i_r is defined, remove the i_r:th variable permanently, and let $D = D - 1$. Go to step 1. Otherwise the algorithm halts.

MATLAB Implementations

The algorithms mentioned in the previous sections are implemented as MATLAB functions. The following conventions have been used in the implementation.

The actual criterion function used is the expression defined in eqs. (9) and (10) plus a term penalizing larger number of input variables. Specifically for classification problems, the criterion used is

$$C'(\mathbf{x}) = C(\mathbf{x}) - p_c D \cdot C(\mathbf{x}) \tag{12}$$

where p_c is a parameter specifying the percentage penalty for having an extra input variable. By default $p_c = 0.005$. For regression problems the criterion used is

$$C'_r(\mathbf{x}) = C_r(\mathbf{x}) - p_r D \cdot C_r(\mathbf{x}) \tag{13}$$

By default $p_r = 0.01$.

A data set is represented by a pair of MATLAB variables Y, X, where X is a $N \times D$ matrix. Each row of X corresponds to a data sample. D is the number of variables and N is the number of samples. Y is a $N \times 1$ vector giving the values of the dependent variable. For classification problems, the value of Y must be specified as in $\{1, 2, 3, ...\}$ corresponding to class categories.

A variable subset is represented by a bit vector, e.g.

$$I = \{1, 1, 0, 1, ...1\}$$

where the i:th bit is set to 1/0 if the i:th variable is included/excluded. This is similar to the binary chromosome representation used in genetic algorithms (GA). Variable subset search can also be formulated as a genetic optimization problem, although this has not been done in this report. Borrowing from the GA jargon, the term "criterion function" is used interchangeably with "fitness function".

The typical sequence of instructions of using this implementation is illustrated in the following.

Some of the MATLAB functions can be speeded up considerably if they are compiled into CMEX codes. This can be done by

```
> kmcompil
```

This is a one-time execution that needs to be done only when the functions are ported to a new machine. MATLAB compiler required.

Assume the data set is stored in a space-delimited format in a text file "yx.data", the first step is loading the data:

```
> load yx.data
```

The data matrix is then partitioned into Y and X part. The following instructions assume that the Y variable is stored in the first column of yx and the rest gives the X variables:

```
> [N,D] = size(yx);
> Y = yx(:,1);
> X = yx(:,2:D);
```

The variables in X may need to be scaled appropriately so they have similar magnitude and variations. For classification problems, the class categories in Y need to be converted to integers between 1 to K, if it is not in conformation with this convention already.

Variable search is invoked by one of the following function calls. For classification problems,

```
    > kmfbclas(Y,X,'forward');          % Forward selection algorithm
Or,
    > kmfbclas(Y,X,'backward');         % Backward elimination algorithm
```

and for regression problems,

```
    > kmfbreg(Y,X,'forward');           % Forward selection algorithm
Or,
    > kmfbreg(Y,X,'backward');          % Backward elimination algorithm
```

Sample scripts demonstrating these steps are given as democlas.m and demoreg.m.

MATLAB is a development tool and as such it is not the best platform for implementing this type of algorithm for application. In the current implementation speed is still a bottleneck. For field applications it is desirable to implement the algorithm in a dedicated program package and utilize the newest generation of powerful processors.

References

[1] Ball, G.H. and D.J. Hall (1965). Isodata, A Novel Method of Data Analysis and Pattern Classification, Stanford Research Institute Technical Report, (NTIS AD699616) Stanford, CA.

[2] Hocking, R.R. (1976). The Analysis and Selection of Variables in Linear Regression, *Biometrics* **32**, 1-49.

[3] Linhart, H. and Zucchini, W. (1986). Model Selection, John Wiley and Sons.

[4] Miller, A.J. (1990). Subset Selection in Regression, Chapman and Hall, London.

[5] Therrien, C.W. (1989). Decision, Estimation, and Classification, John Wiley & Sons, New York.

[6] Thompson, L. (1978). Selection of Variables in Multiple Regression. Part I: A Review and Evaluation, *Int. Stat. Review* **46**, 1-19.
Selection of Variables in Multiple Regression. Part II: Chosen Procedures, Computations and Examples, *Int. Stat. Review* **46**, 129-146 (1978).